For Reference

NOT TO BE TAKEN FROM THIS ROOM

For Reference

NOT TO BE TAKEN FROM THIS ROOM

Ex LIBRIS
UNIVERSITATIS
ALBERTAENSIS

UNIVERSITY OF ALBERTA
QUAECUMQUE VERA

Digitized by the Internet Archive
in 2019 with funding from
University of Alberta Libraries

https://archive.org/details/matrixformulatio00harr

THE UNIVERSITY OF ALBERTA

A MATRIX FORMULATION OF INITIAL VALUE PROBLEMS

by

Harry Maximchuk

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

SEPTEMBER, 1964

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled A MATRIX FORMULATION OF INITIAL VALUE PROBLEMS submitted by Harry Maximchuk in partial fulfilment of the requirements for the degree of Master of Science.

ABSTRACT

This thesis is concerned with numerical approximations to the solutions of initial value problems.

The classical approaches to the problem are reviewed and an algorithm based on Adams' method is presented.

In an attempt to present a unifying theory for general linear multi-step methods, two theorems regarding stability and convergence are proved. The error propagation of multi-step methods is discussed in some detail.

A matrix formulation of initial value problems is given. A factorization procedure is then outlined for a differential equation with nearly constant coefficients. An algorithm for the reduction of the order of a difference approximation when a solution is known is also introduced.

ACKNOWLEDGEMENTS

My gratitude to Professor J. McNamee for, among other things, supervising this thesis, to Dr. D.B. Scott and members of the Department of Computing Science for providing a stimulating atmosphere in which to work and especially to those who helped with this thesis.

TABLE OF CONTENTS

		Page
CHAPTER I	INTRODUCTION	1
Section 1.1	The Initial Value Problem	1
Section 1.2	Definitions and Notation	3
Section 1.3	A Survey of the Thesis	10
CHAPTER II	CLASSICAL METHODS FOR SOLVING THE INITIAL VALUE PROBLEM	12
Section 2.1	Introduction	12
Section 2.2	The Taylor Series Method	13
Section 2.3	The Method of Undetermined Coefficients	14
Section 2.4	The Lagrangian Polynomial	19
Section 2.5	Adams-Bashforth Method	23
Section 2.6	Adams-Moulton Method	26
Section 2.7	Nyström's Method	30
Section 2.8	The Generalized Milne-Simpson Method	31
Section 2.9	The Hermite Interpolation Formula	32
Section 2.10	Gaussian Quadrature	35
CHAPTER III	LINEAR MULTI-STEP METHODS	38
Section 3.1	Introduction	38
Section 3.2	Existence and Uniqueness of the Solution of the Difference Equation	40
Section 3.3	Condition for Stability	42
Section 3.4	Condition for Convergence	46
Section 3.5	Propagation of Errors	51
Section 3.6	Optimizing the Difference Operator	54

		Page
CHAPTER IV	A MATRIX FORMULATION OF THE INITIAL VALUE PROBLEM	55
Section 4.1	Introduction	55
Section 4.2	Matrix Formulation of a Linear Initial Value Problem	57
Section 4.3	Solution of Linear Equations With Nearly Constant Coefficients	60
Section 4.4	The Reduction of the Order of a Difference Approximation when a Solution is Known	67
CHAPTER V	CONCLUDING REMARKS	69
BIBLIOGRAPHY		71
APPENDIX		74

LIST OF TABLES

		Page
TABLE I	Solution of $y''' - 7y' + 6y = 0$	66
TABLE II	Solution of $y'' - (1 - \frac{1}{4x^2})y = 0$	76
TABLE III	Comparison of Solutions	77

CHAPTER I

Introduction

One aim of this thesis is to review the general linear multi-step methods used in solving initial value problems. A second aim is to explore methods for finding all the solutions of an nth order initial value problem.

The introductory chapter gives a statement of the problem; defines terms and establishes a notation: finally, the contents of each chapter are summarized.

Section 1.1 The Initial Value Problem

The theory of differential equations is in essence the following problem - given a differential equation of order n, i.e., an equation of the form

$$F(x, y, y', y'', \ldots, y^{(n)}) = 0 \quad , \tag{1.1.1}$$

it is required to find a solution $y(x)$ satisfying (1.1.1). As is well known from elementary examples, a given differential equation may have many solutions. In order to pin down a solution of a given differential equation, it is necessary to specify some additional properties, such as its value and the values of some of its derivatives at specified points. It is well-known that for an equation of order n a minimum of n conditions are needed to fix a solution. If the n conditions are specified at one point, then the differential equation plus the n conditions form an initial value problem.

Throughout this thesis we assume that the differential equation is well-posed, that is,

1. The solution must exist.
2. The solution must be uniquely determined.
3. The solution must depend continuously on the data of the problem.

In differential systems arising from problems in mathematical physics, this is generally true.

Throughout the years, various techniques have been developed to solve certain classes of problems. But very often the explicit solution given by traditional techniques may be tedious to evaluate. For instance,

$$y(x) = e^{-x^2} \int_0^x e^{t^2}\, dt$$

is the solution of the differential equation

$$y' = 1 - 2xy .$$

In order to determine numerical values of $y(x)$, one has to calculate an integral that is not expressible in terms of elementary functions and that is not adequately tabulated. Thus numerical methods are used extensively in solving differential systems.

In essence, the numerical solution of a differential system is simple to describe: the differential equation and its accompanying differential conditions are converted to a difference

equation and its accompanying difference conditions. The difference problem is then susceptible of a numerical solution at discrete points in the region of interest. However, the difference problem, at times, may not be well-posed.

Before proceeding any further, it is necessary to define pertinent terms and to establish a notation. This is done in the following section.

Section 1.2 Definitions and Notation

As a preliminary to this section, we first describe the reduction of higher-order differential equations subject to initial conditions, to a set of first order equations.

Although the most general form of an nth order ordinary differential equation is that of (1.1.1), we consider a slightly restricted form, namely,

$$(1.2.1) \quad y^{(n)}(x) = f(y^{(n-1)}(x), y^{(n-2)}(x), \ldots, y'(x), y(x), x).$$

That is, we assume that we can solve (1.1.1) formally for the highest derivative present.

Let us suppose that (1.2.1) is subject to the initial conditions

$$y(x_o) = \eta_o$$

$$y'(x_o) = \eta_1$$

$$\vdots \qquad \vdots$$

(1.2.2) $$y^{(n)}(x_o) = \eta_n$$

at $x = x_o$.

Also we can write

$$y_o(x) = y(x)$$

$$y_1(x) = y'(x)$$

$$y_2(x) = y''(x)$$

$$\vdots \qquad \vdots$$

(1.2.3) $$y_n(x) = y^{(n)}(x) = f(y^{(n-1)}(x), y^{(n-2)}(x), \ldots, y'(x), y(x), x).$$

Differentiating the first of (1.2.3), we have

$$y_o' = y' = y_1$$

That is,

(1.2.4) $$y_o' = y_1$$

Similarly, $$y_1' = y_2$$

$$y_2' = y_3$$

$$\vdots \qquad \vdots$$

$$(1.2.5)\qquad y_{n-1}' = y_n = f(y_{n-1}, y_{n-2}, \ldots, y_o, x)$$

The initial conditions (1.2.2) can be written as

$$y_o(x_o) = \eta_o$$

$$y_1(x_o) = \eta_1$$

$$\vdots \qquad \vdots$$

$$(1.2.6)\qquad y_n(x_o) = \eta_n$$

Or, using a vector notation

$$(1.2.7)\qquad \underline{y}(x_o) = \underline{\eta}$$

It is clear that the system

$$y_o'(x) = f_o(x, y_o, y_1, \ldots, y_n)$$

$$y_1'(x) = f_1(x, y_o, y_1, \ldots, y_n)$$

$$\vdots \qquad \vdots$$

$$(1.2.8)\qquad y_n'(x) = f_n(x, y_o, y_1, \ldots, y_n)$$

or more precisely,

$$(1.2.9)\qquad \underline{y}'(x) = \underline{f}(x, \underline{y})$$

subject to the initial conditions (1.2.7) is in effect the differential equation (1.2.1) subject to the initial conditions (1.2.2).

We assume that $\underline{f}(x,\underline{y})$ belongs to a class of functions that satisfy a Lipschitz condition of order one.

Definition 1.2.1

A continuous function $f_i(x,\underline{y})$ satisfies a Lipschitz condition of order α with respect to $\underline{y}$ in a domain $\overline{D}$ if there exists a positive constant L such that

$$(1.2.10) \quad | f_i(x,\underline{Y}) - f_i(x,\underline{y}) | < L \sum_{i=o}^{n} | Y_i - y_i |^{\alpha}$$

for $(x,\underline{Y})$ and $(x,\underline{y})$ in the domain D.

The Lipschitz condition implies continuity with respect to $\underline{y}$ and is implied by differentiability with respect to $\underline{y}$. A bound on the error of approximation is determined in Chapter III in terms of the Lipschitz constant, L.

As was stated in the previous section, one of the first problems in solving a differential equation is to replace it by a difference operator.

Definition 1.2.2

The general k-step difference operator may be written as

$$(1.2.11) \quad L[\underline{y}(x)] = \sum_{q=o}^{k} \alpha_q \underline{y}_{n+q} - h \beta_q \underline{y}'_{n+q} = 0$$

where

$$\underline{y}_{n+q} = \underline{y}(x_{n+q})$$

$$= \underline{y}(x_n + qh) \ .$$

If $\beta_k = 0$, the difference operator is explicit. That is, y_{n+k} can be solved for explicitly and can be used as a first approximation. Thus the operator is a predictor for y_{n+k}. However, if $\beta_k \neq 0$, the operator is implicit. By knowing an approximation for y_{n+k}, a more accurate value of y_{n+k} can be calculated using the implicit operator. Hence, it is a corrector for y_{n+k}.

An algorithm using the Adams-Bashforth method as a predictor and the Adams-Moulton method as a corrector illustrates in Chapter II an efficient procedure for determining discrete values of $y(x)$ step by step.

In order that the operator (1.2.11) be a useful method for integrating a differential equation, it must possess certain properties. First of all, it must be stable.

Definition 1.2.3

A method is stable if any bounded starting procedure yields a uniformly bounded solution of the difference equation as $h \to 0$.

Theorem 3.3.1 in Chapter III prescribes the necessary conditions for stability of linear multi-step methods.

Stability does not imply convergence of an approximation, as $h \to 0$, to a solution of the original problem. To illustrate this fact, we consider the problem $y' = 0$, $y(0) = 1$ and use the

ridiculous but nevertheless stable numerical method defined by $y_o = 1$ and $y_{n+1} = \frac{1}{2} y_n$. However, stability does ensure a good global behaviour of a numerical method.

Some other condition is needed to ensure a good local behaviour of the method. We call this condition the consistency condition.

Definition 1.2.4

If $h = \max (x_{q+1} - x_q)$

then the operator (1.2.11) is consistent with the problem (1.2.9) if (1.2.11) is satisfied by any solution of (1.2.9), except for terms which are o(h). Or more precisely, if $\underline{y}(x)$ is any solution of (1.2.9), then by substituting

$$\underline{y}(x_n+qh) = \underline{y}(x_n) + qh\, \underline{y}'(x_n) + o(h) \quad \text{for} \quad \underline{y}_{n+q}$$

and $$\underline{y}'(x_n+qh) = \underline{y}'(x_n) + o(1) \quad \text{for} \quad \underline{y}'_{n+q}$$

in (1.2.11) and equating coefficients, we obtain

$$\sum_{q=o}^{k} \alpha_q = o$$

$$\sum_{q=o}^{k} q\, \alpha_q = \sum_{q=o}^{k} \beta_q \qquad (1.2.12)$$

which we call the consistency conditions.

Definition 1.2.5

A method is convergent if the values of the solution of the approximating difference equation tend to the values of the exact solution as $h \rightarrow 0$.

Theorem 3.4.1 shows that a multi-step method is convergent if and only if it is both stable and consistent.

The analysis of the errors involved in the approximation by a difference system of the differential system and of the errors accrued while solving the difference system is extremely difficult and far from complete. Two points of view are available for the analysis.

Definition 1.2.6

The a priori approach assumes that the true analytic solution $\underline{y}_T$, is available. When the approximate solution, $\underline{y}_A$, is calculated, the a priori error may be defined as $\underline{y}_T - \underline{y}_A$.

Since very often the analytic solution is not available, this approach to error analysis is fruitless.

Definition 1.2.7

The a posteriori approach attempts to improve an approximation $\underline{y}_A$. If the improved approximation is $\underline{y}_{AA}$, the a posteriori error may be defined as $\underline{y}_{AA} - \underline{y}_A$.

The latter approach is used in this thesis. However, to be of any practical use, the true solution should be bounded in some fashion by the a posteriori error.

There are two types of errors that arise in a multi-step method.

Definition 1.2.8

The discretization error, E_n, is uniquely determined by the initial value problem, the starting procedure, the continuation method and the step h.

Definition 1.2.9

The round-off error, R_n, depends on the method, the differential system, the computing machine used, the order in which arithmetic operations are performed and other details of the algorithm.

The asymptotic behaviour of both the discretization error and the round-off error is discussed in Chapter III.

The notation used in the programs in this thesis is a modification of an algorithmic language proposed by Iverson [1962].

Section 1.3 A Survey of the Thesis

This thesis attempts to review and analyse critically some major techniques developed for the numerical solution of the initial value problem.

Chapter II reviews the more classical approaches to the solution of the initial value problem such as a truncated Taylor series, the method of undetermined coefficients, Lagrangian interpolation and Gaussian quadrature.

In each case, the sense of the approximation is evaluated.

In Chapter III, the general multi-step methods are developed and their stability, consistency, and convergence properties are discussed in some detail. The asymptotic behaviour of the discretization and the round-off errors is also discussed. By modifying the difference operator, the results are then extended to non-equidistant points in an attempt to optimize the operator.

In Chapter IV, we consider the matrix formulation of an nth order differential equation by a finite-difference approximation. An algorithm is developed for the elimination of the undesired solutions of the difference equation and a step-by-step method generates a solution.

The final chapter summarizes the major results of the thesis, and indicates some topics on which current research is proceeding.

Each chapter is preceded by a brief synopsis.

CHAPTER II

CLASSICAL METHODS OF SOLVING THE INITIAL VALUE PROBLEM

This chapter discusses some of the classical techniques for the solution of the initial value problem.

The Taylor series method of approximation and a method of undetermined coefficients are developed for sets of first order initial value differential equations.

The more practical procedures, such as Adams-Bashforth, Adams-Moulton, Nyström, and Milne-Simpson, are discussed in detail. An algorithm based on the first two methods is presented.

Integration methods based on the orthogonal polynomials are also outlined.

Section 2.1 Introduction

Most of the classical methods of solving numerically the initial value problem are based on the method of interpolation. That is, we have at our disposal a certain portion of the curve which can be interpolated with sufficient accuracy by a polynomial of not too high order, provided the step-size h is chosen small enough. This polynomial can now be applied for extrapolating to the next point x_{k+1}. The procedure is then repeated by including this new point and dropping the extreme point of the previous step. In this fashion there is always the same length of curve under the search light, while this light moves slowly forward to newer regions, until the entire range of x is exhausted.

Our aim in this chapter is to exhibit some of these polynomials; the methods described are not necessarily the

methods to be used in everyday practice.

Section 2.2 The Taylor Series Method

Let us consider the system of equations (1.2.9),

$$\underline{y}'(x) = \underline{f}\,(x,\underline{y}) \tag{2.2.1}$$

Differentiating with respect to x,

$$\underline{y}''(x) = \frac{\partial \underline{f}}{\partial x} + \frac{\partial \underline{f}}{\partial \underline{y}}\,\underline{y}' \tag{2.2.2}$$

Assuming sufficient differentiability of $\underline{f}$ with respect to all its arguments, we can obtain expressions for derivatives of $\underline{y}(x)$ of all orders. This is done recursively. Using (2.2.2), we obtain $\underline{y}''(x)$ at $x = x_o$. These values can then be substituted into the expression for $\underline{y}'''(x)$, obtaining $\underline{y}'''(x_o)$, and so forth, for as many terms as we need, or can reasonably obtain.

We can now write $\underline{y}(x)$ as a Taylor series; in particular for $x = x_o + h$,

$$\underline{y}(x_o + h) = \underline{y}(x_o) + h\,\underline{y}'(x_o) + \frac{h^2}{2!}\,\underline{y}''(x_o) + \ldots \tag{2.2.3}$$

Since $\underline{y}(x_o)$ are the given initial conditions (1.2.7) and the values of all the required derivatives of $\underline{y}(x_o)$ are calculable as described above, we obtain values for $\underline{y}(x_o + h)$.

In practice, we do not use the infinite series (2.2.3). Since higher-order derivatives are often quite

difficult to evaluate, we choose h small enough to guarantee convergence to the desired number of figures, using only those derivatives which are evaluable with reasonable effort.

Once $\underline{f}(x)$ has been advanced to $x = x_o + h$, we can regard it as new "initial conditions", and advance to the next point, usually $x = x_o + 2h$.

The Taylor series method, as described above, has advantages and disadvantages over other methods discussed in this chapter. In its favour is the fact that it is applicable to a very wide class of problems, including non-linear systems. Also, it requires no special starting procedure. On the other hand, the evaluation of the necessary derivatives may be difficult since automatic digital computer programs are unable to evaluate the derivatives of an arbitrary function, given only its form; that is, without formulae for these derivatives being worked out by hand. Furthermore, in determining the derivatives, there are more functions to evaluate which is time-consuming. Hence, we are faced with the problem of optimizing; that is, we include as many derivatives as are available and choose h small enough to guarantee convergence.

Section 2.3 The Method of Undetermined Coefficients

The main drawback of the Taylor series method described in the previous section is finding all the required derivatives explicitly. In this section we develop a procedure

based on local Taylor expansions which overcomes this problem. Although we still consider the system (1.2.9), the procedure described in this section is applied to each equation individually. Hence let us consider a typical equation,

$$y' = f(x,\underline{y}) \tag{2.3.1}$$

Let us assume that by some extrapolation method we arrived at a certain point $x = kh$. That is, we have at our disposal the values $y_k = y(hk)$. By substituting these into (2.3.1), we can evaluate the derivatives

$$y_k' = y'(kh).$$

Also let us assume that we have at our disposal 2 m data

$$\begin{aligned} z_o &= y(x - mh) & z_o' &= y'(x - mh) \\ z_1 &= y(x - (m-1)h) & z_1' &= y'(x - (m-1)h) \\ &\vdots & &\vdots \\ z_{m-1} &= y(x-h) \quad , & z_{m-1}' &= y'(x-h) \end{aligned} \tag{2.3.2}$$

We now choose as our point of reference the point x into which we want to extrapolate. Hence we want to determine $z_m = y(x)$ on the basis of our data. For this

purpose we expand $y(x)$ in the vicinity of the point x in a Taylor series, leaving the number of terms of the expansion undetermined:

$$z_{m-1} = y(x-h) = y(x) - hy'(x) + \frac{h^2}{2!} y''(x) - \frac{h^3}{3!} y'''(x) + \ldots$$

$$z_{m-2} = y(x-2h) = y(x) - 2hy'(x) + \frac{4h^2}{2!} y''(x) - \frac{8h^3}{3!} y'''(x) + \ldots$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$(2.3.3) \qquad z_o = y(x-mh) = y(x) - mh\, y'(x) + \frac{m^2h^2}{2!} y''(x) - \frac{m^3h^3}{3!} y'''(x) + \ldots$$

To these expressions we will add the expansion of the first derivatives:

$$-h\, z'_{m-1} = -h\, y'(x-h) = -h\, y'(x) + h^2 y''(x) - \frac{h^3}{2!} y'''(x) + \ldots$$

$$-h\, z'_{m-2} = -h\, y'(x-2h) = -h\, y'(x) + 2h^2 y''(x) - \frac{4h^3}{2!} y'''(x) + \ldots$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$(2.3.4) \quad -h\, z_o' = -h\, y'(x-mh) = -h\, y'(x) + mh^2 y''(x) - \frac{m^2 h^3}{2!} y'''(x) + \ldots$$

Now if we multiply equations (2.3.3) by some undetermined factors $\alpha_1, \alpha_2, \ldots, \alpha_m$ and equations (2.3.4) by $-\beta_1, -\beta_2, \ldots, -\beta_m$ and add these equations, the left side becomes

$$(2.3.5) \quad \alpha_1 z_{m-1} + \alpha_2 z_{m-2} + \ldots + \alpha_m z_o + h(\beta_1 z_{m-1}' + \beta_2 z_{m-2}' + \ldots + \beta_m z_o')$$

On the right side the factor of $y(x)$ becomes

$$(2.3.6) \quad \alpha_1 + \alpha_2 + \ldots + \alpha_m$$

This factor we want to make equal to 1 because our approximation is to predict $y(x)$ (that is, z_m) in terms of all previous z_{m-1}:

$$(2.3.7) \quad z_m = \sum_{k=1}^{m} \alpha_k z_{m-k} + h \sum_{k=1}^{m} \beta_k z_{m-k}'$$

Our program cannot be carried out with absolute accuracy but it can be accomplished with a high degree of accuracy if we succeed in obliterating on the right side the

factors of h, h^2, etc. We have 2m coefficients at our disposal and thus 2m degrees of freedom. One degree is absorbed by making the sum (2.3.6) equal to 1. The remaining 2m-1 degrees of freedom can be used to obliterate on the right side the powers of h, up to order h^{2m-1}. Hence we shall obtain the extrapolated value z_m with an error proportional to h^{2m}.

The linear system of equations obtained for the determination of the coefficients α_i, β_i is given as follows:

$$\alpha_1 + \alpha_2 + \ldots + \alpha_m = 1$$

$$\alpha_1 + 2\alpha_2 + \ldots + m\alpha_m - (\beta_1 + \beta_2 + \ldots + \beta_m) = 0$$

$$\alpha_1 + 4\alpha_2 + \ldots + m^2\alpha_m - 2(\beta_1 + 2\beta_2 + \ldots + m\beta_m) = 0$$

$$\vdots$$

$$\alpha_1 + 2^{2m-1}\alpha_2 + \ldots + m^{2m-1}\alpha_m - (2m-1)$$

$$(\beta_1 + 2^{2m-2}\beta_2 + \ldots + m^{2m-2}\beta_m) = 0 \qquad (2.3.8)$$

Obviously, the method just described requires the solution of 2m simultaneous equations which is time-consuming if m is large. Another disadvantage of the above procedure and of the predictor-corrector techniques discussed in the following sections is that they are not self-starting. That is, a starting

procedure is required to obtain the 2m data (2.3.2). In practice, we could use a Taylor series, Euler's method with very small h, or a Runge-Kutta method to get the solution going.

Section 2.4 The Lagrangian Interpolation Polynomial

Instead of solving the linear algebraic system (2.3.8) for coefficients α_i, β_i, we follow a somewhat different line of approach which sheds new light on the nature of our approximation.

Let the function f(x) be defined at q+1 distinct points $x_o, x_1, \ldots, x_q$. It is well known that among all the polynomials in x of degree not exceeding q there exists exactly one polynomial P(x) which satisfies the relations

$$P(x_\nu) = f(x_\nu) \qquad \nu = o, 1, \ldots, q \tag{2.4.1}$$

or which interpolates the function f(x) at the points x_ν. This polynomial is of the form:

$$P(x) = \pi(x) \sum_{\nu=o}^{q} \frac{f(x_\nu)}{\pi'(x_\nu)(x-x_\nu)} \tag{2.4.2}$$

where

$$\pi(x) = (x-x_o)(x-x_1) \ldots (x-x_q). \tag{2.4.3}$$

It is natural to expect that $P(x)$ reproduces $f(x)$ not only exactly at the points $x = x_\nu$, but also approximately for other values of x. The following lemma estimates the error committed in this approximation.

Lemma 2.4.1

Let $f(x)$ have a continuous derivative of order $q+1$ in an interval J. Then for every point x in J there exists a point ξ in an interval containing x and x_ν ($\nu = 0,1, \ldots, q$) such that

$$f(x) - P(x) = \frac{1}{(q+1)!}\,\pi(x)\,f^{(q+1)}(\xi). \tag{2.4.4}$$

Proof:

If x is one of the points x_ν, both sides of (2.4.4) vanish for arbitrary ξ. If $x \neq x_\nu$, $\nu = 0,1,\ldots, q$, consider the auxiliary function of the variable t,

$$F(t) = f(t) - P(t) - \lambda\,\pi(t) \tag{2.4.5}$$

where

$$\lambda = \frac{f(x) - P(x)}{\pi(x)} \tag{2.4.6}$$

so that $F(x) = 0$. Obviously, the function $F(t)$ also vanishes at all the points x_ν, thus vanishing at $q+2$ points. By repeated application of Rolle's theorem, its μth derivative vanishes at least at $q+2-\mu$ distinct points ($\mu = 1,2,\ldots, q+1$). Let ξ be a point where the $(q+1)^{st}$ derivative vanishes. Since

$P^{(q+1)}(\xi) = 0$ and $\pi^{(q+1)}(\xi) = (q+1)!$, we have at this point

$$(2.4.7) \qquad F^{(q+1)}(\xi) = f^{(q+1)}(\xi) - \lambda(q+1)! = 0$$

Using (2.4.6) and (2.4.7), we get the result

$$f(x) - P(x) = \frac{1}{(q+1)!}\,\pi(x)\, f^{(q+1)}(\xi).$$

In applications to differential equations the points x_ν are usually equally spaced. The interpolating polynomial is then most conveniently represented in terms of finite differences. We assume that $x_p = x_o + ph$, where h is a constant. Writing $f_p = f(x_p)$, the first backward difference of the function $f(x)$ at the point $x = x_p$ is defined as

$$(2.4.8) \qquad \nabla f_p = f_p - f_{p-1}$$

Higher backward differences are defined by

$$(2.4.9) \qquad \nabla^q f_p = \nabla(\nabla^{q-1} f_p).$$

If one puts $\nabla^o f_p = f_p$, it is clear that

$$(2.4.10) \qquad \nabla^q f_p = \sum_{m=o}^{q} (-1)^m \binom{q}{m} f_{p-m}, \quad q = o,1,\ldots$$

where $\binom{q}{m}$ is the binomial coefficient

$$\binom{q}{m} = \frac{q(q-1)\ \ldots\ (q-m+1)}{1(2)\ \ldots\ (m)} \qquad m = 1,2,\ldots$$

$$(2.4.11) \qquad \binom{q}{o} = 1.$$

Formula (2.4.10) expresses the differences in terms of function values or ordinates. It is also helpful to express the ordinates in terms of differences. It can be verified by induction that

$$f_{p-q} = \sum_{m=o}^{q} (-1)^m \binom{q}{m} \nabla^m f_p \quad q = o,1, \ldots \tag{2.4.12}$$

In order to express the interpolating polynomial in terms of differences, let us introduce the variable

$$s = \frac{x - x_p}{h} \tag{2.4.13}$$

The polynomial $P(x)$ of degree $\leq q$ which interpolates the function $f(x)$ at the points x_p, x_{p-1}, ..., x_{p-q} can be written as

$$P(x) = \sum_{m=o}^{q} (-1)^m \binom{-s}{m} \nabla^m f_p \, . \tag{2.4.14}$$

Since $\binom{-s}{m}$ is a polynomial of degree m in s and since s is a linear function of x, this is indeed a polynomial of degree of $\leq q$. Also for $r = o,1, \ldots, q$

$$P(x_{p-r}) = \sum_{m=o}^{q} (-1)^m \binom{r}{m} \nabla^m f_p$$

$$= \sum_{m=o}^{r} (-1)^m \binom{r}{m} \nabla^m f_p$$

(2.4.15) $$= f_{p-r}$$

Since $\binom{r}{m} = 0$ for r an integer $< m$.

Section 2.5 Adams - Bashforth Method

An exact solution of any of the first-order differential equations (1.2.9) satisfies the identity

(2.5.1) $$y(x+k) - y(x) = \int_x^{x+k} f(t, y(t))\, dt$$

for any points x and x+k. The methods now to be discussed are based on replacing the $f(x,y(x))$ by an interpolating polynomial having the values $f_n = f(x_n, y_n)$ on a set of points x_n. We assume that the interpolating points are $x_p, x_{p-1}, \ldots, x_{p-q}$; the polynomial replacing $f(x, y(x))$ is given by (2.4.14). Thus, the Adams-Bashforth formula becomes

$$y_{p+1} - y_p = \int_{x_p}^{x_{p+1}} P(x)\, dx$$

(2.5.2) $$= h \sum_{m=o}^{q} \gamma_m \nabla^m f_p$$

where $$\gamma_m = (-1)^m \frac{1}{h} \int_{x_p}^{x_{p+1}} \binom{-s}{m}\, dx$$

$$(2.5.3) \qquad = (-1)^m \int_0^1 \binom{-s}{m} ds$$

In order to calculate the γ_m explicitly, we use the method of generating functions. Let

$$G(t) = \sum_{m=0}^{\infty} \gamma_m t^m$$

$$= \int_0^1 \sum_{m=0}^{\infty} (-t)^m \binom{-s}{m} ds$$

$$= \int_0^1 (1-t)^{-s} ds$$

$$(2.5.4) \qquad = - \frac{t}{(1-t) \log (1-t)}$$

Then (2.5.4) can be written as

$$- \frac{\log (1-t)}{t} G(t) = \frac{1}{(1-t)}$$

Using the expansions

$$\frac{1}{1-t} = 1 + t + t^2 + \ldots$$

$$(2.5.5) \qquad - \frac{\log(1-t)}{t} = 1 + \frac{1}{2}t + \frac{1}{3}t^2 + \ldots$$

we obtain the identity

$$(2.5.6)\quad (1 + \tfrac{1}{2}t + \tfrac{1}{3}t^2 + \ldots)(\gamma_o + \gamma_1 t + \gamma_2 t^2 + \ldots) = 1 + t + t^2 + \ldots$$

By comparing the coefficients of corresponding powers of t, we obtain the successive values

m	0	1	2	3	4	5	6
γ_m	1	$\frac{1}{2}$	$\frac{5}{12}$	$\frac{3}{8}$	$\frac{251}{720}$	$\frac{95}{288}$	$\frac{19087}{60480}$

The explicit use of the differences $\nabla^m f_p$ is especially suitable for hand computation. For instance, if the difference of the highest order $\nabla^q f_p$ is large, this may be an indication that the accuracy is insufficient and that the number of terms in (2.5.2) should be increased. However, if the number q is fixed as is frequently the case when Adams method is programmed for an automatic computer, there is no particular reason why the difference should be carried explicitly.

If we now write the Adams-Bashforth formula explicitly with a slight change of notation, we get

$$(2.5.7)\quad y_{p+1} - y_p = h(1 + \tfrac{1}{2}\nabla + \tfrac{5}{12}\nabla^2 + \tfrac{3}{8}\nabla^3 + \ldots)y'_p \quad .$$

Suppose that we truncate the series after the second difference. Since we include the term y'_{p-2} , our equation is

$$(2.5.8) \qquad y_{p+1} - y_p = \frac{h}{12}(23y'_p - 16y'_{p-1} + 5y'_{p-2}).$$

Clearly, the highest-order derivative occurring is y'_p. Hence, the formula is explicit and can be used as a predictor for y_{p+1}. In the following section we develop an analogous corrector formula.

Section 2.6 Adams - Moulton Method

Let us now consider

$$y_p - y_{p-1} = \int_{x_{p-1}}^{x_p} P(x)\, dx$$

$$(2.6.1) \qquad = h \sum_{m=o}^{q} \gamma_m^* \nabla^m f_m$$

where $$\gamma_m^* = \frac{(-1)^m}{h} \int_{x_{p-1}}^{x_p} \binom{-s}{m} dx$$

$$(2.6.2) \qquad = (-1)^m \int_{-1}^{o} \binom{-s}{m} ds$$

By operations analogous to those of section 2.5, the generating function is

$$G^*(t) = -\frac{t}{\log(1-t)} \tag{2.6.3}$$

That is,

$$-\frac{\log(1-t)}{t} = 1$$

Hence, by using (2.5.5)

$$(1 + \tfrac{1}{2}t + \tfrac{1}{3}t^2 + \ldots)(\gamma_o^* + \gamma_1^* t + \ldots) = 1 \tag{2.6.4}$$

Or numerically, the values are as follows:

m	0	1	2	3	4	5	6
γ_m^*	1	$-\frac{1}{2}$	$-\frac{1}{12}$	$-\frac{1}{24}$	$-\frac{19}{720}$	$-\frac{3}{160}$	$-\frac{863}{60480}$

It is interesting to note that for q = 1 the Adams-Moulton formula is actually the trapezoidal rule.

If we write the Adams-Moulton formula explicitly, we get

$$y_p - y_{p-1} = h(1 - \tfrac{1}{2}\nabla - \tfrac{1}{12}\nabla^2 - \tfrac{1}{24}\nabla^3 - \ldots)y'_p \,. \tag{2.6.5}$$

Again, truncating after the second difference, we get

$$(2.6.6) \qquad y_p - y_{p-1} = \frac{h}{12}(5y'_p + 8y'_{p-1} - y'_{p-2})$$

Clearly, the formula is implicit and may be used as a corrector. An algorithm using the Adams-Bashforth method as a predictor and the Adams-Moulton method as a corrector follows.

At this point, we return to the original set of first-order equations (1.2.9) subject to (1.2.7). Let us represent the vectors $\underline{y}_{n-3}$, $\underline{y}_{n-2}$, $\underline{y}_{n-1}$ and $\underline{y}_n$ as the columns $\underline{Y}_1$, $\underline{Y}_2$, $\underline{Y}_3$ and $\underline{Y}_4$ respectively of a matrix $\underline{Y}$. The vectors $\underline{y}'_{n-3}$, $\underline{y}'_{n-2}$, $\underline{y}'_{n-1}$ and $\underline{y}'_n$ we represent as the columns $\underline{W}_1$, $\underline{W}_2$, $\underline{W}_3$ and $\underline{W}_4$ of a matrix $\underline{W}$. $\underline{\varepsilon}$ is a unit vector. We assume that some starting process has calculated and output the values of $\underline{Y}_1$, $\underline{Y}_2$, $\underline{Y}_3$, $\underline{W}_1$, $\underline{W}_2$, $\underline{W}_3$ for $x = x_o$, $x_o + h$, $x_o + 2h$ respectively. The vector $\underline{Y}_4$ is predicted from (2.5.8) and corrected by using (2.6.6). d is the tolerance that must be met by iteration.

ADAMS METHOD

1	$x \leftarrow x_o + 3h$	initialize x
2	$\underline{Y}_4 \leftarrow \underline{Y}_3 + \frac{h}{12}(23\underline{W}_3 - 16\underline{W}_2 + 5\underline{W}_1)$	calculate predictors
3	$\underline{W}_4 \leftarrow \underline{f}(x, \underline{Y}_4)$	calculate derivatives
4	$\underline{Y}_4 \leftarrow \underline{Y}_3 + \frac{h}{12}(5\underline{W}_4 + 8\underline{W}_3 - \underline{W}_2)$	calculate correctors

5	$\underline{n} \leftarrow \underline{f}\ (x, \underline{Y}_4)$	calculate modified derivatives
6	$\underline{m} \leftarrow \frac{5h}{12}(\underline{n} - \underline{W}_4)$	calculate modifiers
7	$\underline{W}_4 \leftarrow \underline{n}$	
8	$\underline{Y}_4 \leftarrow \underline{Y}_4 + \underline{m}$	modify correctors
9	$\lvert\underline{m}\rvert : d\ \underline{\varepsilon}\ ,\ (\leq,>) \rightarrow (10,5)$	test for tolerance
10	$\underline{Y}_1 \leftarrow \underline{Y}_2$	Shift columns
11	$\underline{Y}_2 \leftarrow \underline{Y}_3$	of $\underline{Y}$ left
12	$\underline{Y}_3 \leftarrow \underline{Y}_4$	one position
13	$\underline{W}_1 \leftarrow \underline{W}_2$	Shift columns
14	$\underline{W}_2 \leftarrow \underline{W}_3$	of $\underline{W}$ left
15	$\underline{W}_3 \leftarrow \underline{W}_4$	one position
16	PRINT : x, $\underline{Y}_4$	output results
17	$x \leftarrow x + h$	increment x
18	$x : x_{max}$, $(\leq,>) \rightarrow (2,19)$	test for final x
19	STOP	

Inspection of this program shows that we are carrying four columns each of $\underline{Y}$ and $\underline{W}$ and one each of $\underline{m}$ and $\underline{n}$, for a total of 10ℓ computer storage locations, where ℓ is the number of simultaneous equations being solved. The routine should be fairly fast, the major time being spent in the evaluations of $\underline{f}$.

Section 2.7 Nyström's Method

Here we set

$$y_{p+1} - y_{p-1} = \int_{x_{p-1}}^{x_{p+1}} P(x)\,dx \tag{2.7.1}$$

$$= h \sum_{m=o}^{q} \kappa_m \nabla^m f_p$$

where

$$\kappa_m = (-1)^m \frac{1}{h} \int_{x_{p-1}}^{x_{p+1}} \binom{-s}{m} dx$$

$$= (-1)^m \int_{-1}^{1} \binom{-s}{m} ds \tag{2.7.2}$$

Obviously, this method is explicit and works much like the Adams-Bashforth method, except that now the increment of y_n is calculated over two steps in place of one.

The generating function of the coefficients κ_m is found to be

$$K(t) = - \frac{t}{\log(1-t)} \frac{2-t}{1-t} \tag{2.7.3}$$

which results in the following values:

m	0	1	2	3	4	5	6
κ_m	2	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{29}{90}$	$\frac{14}{45}$	$\frac{1139}{3780}$.

The special case $q = 0$ of Nyström's method is called the mid-point rule. Since $\kappa_1 = 0$, it yields the same accuracy as could ordinarily be expected from taking $q = 1$, which accounts for some of the popularity of the method.

Section 2.8 The Generalized Milne-Simpson Method

This results in taking $x = x_{p-2}$, $k = 2h$ in (2.5.1)

That is,

$$y_p - y_{p-2} = h \int_{x_{p-2}}^{x_p} P(x)\, dx$$

$$= h \sum_{m=o}^{q} \kappa_m^* \nabla^m f_p \tag{2.8.1}$$

where

$$\kappa_m^* = (-1)^m \frac{1}{h} \int_{x_{p-2}}^{x_p} \binom{-s}{m} dx$$

$$= (-1)^m \int_{-2}^{o} \binom{-s}{m} ds \tag{2.8.2}$$

Formula (2.8.1) resembles the Adams-Moulton formula in being implicit. Although the integration is over two steps, for $q \geq 2$ the method is more accurate than any of the methods previously considered. This is particularly true in the case $q = 2$, which has been outlined by Milne [1953].

The generating function is given by

$$K^*(t) = - \frac{t}{\log(1-t)} \quad (2-t) \tag{2.8.3}$$

which results in the values:

m	0	1	2	3	4	5	6
κ_m^*	2	-2	$-\frac{1}{3}$	0	$-\frac{1}{90}$	$-\frac{1}{90}$	$-\frac{37}{3780}$

We note that $\kappa_3^* = 0$, which shows that the Milne-Simpson formula with q = 2 produces an effect which normally would be expected only with q = 3. By taking q = 2 and writing Milne's formula in terms of ordinates, we get

$$y_p - y_{p-2} = \frac{1}{3} h(f_p + 4f_{p-1} + f_{p-2}) \tag{2.8.4}$$

which is the familiar Simpson formula in the theory of numerical integration. It is obvious that Nyström's method combined with Milne-Simpson's method forms another predictor-corrector procedure.

Section 2.9 Hermite's Interpolation Formula

In the previous sections, we assumed that our data points were equally spaced. We now take a more general point of view. That is, given 2n data y_i and y_i', i = 1,2, . . . , n at distinct points x_i, we want to find a polynomial of order 2n-1 which satisfies the data.

Let us consider the polynomial (2.4.3) with q = n, namely,

$$\pi(x) = (x-x_1)(x-x_2) \ldots (x-x_n). \tag{2.9.1}$$

If

$$\ell_i(x) = \frac{\pi(x)}{(x-x_i)\,\pi'(x_i)}\ , \tag{2.9.2}$$

clearly $\ell_i(x_j) = \delta_{ij}$

where δ_{ij} is the Kronecker delta function. Then $\ell_i^2(x)$ which is of order 2n-2, has the property that it vanishes along with its derivative at $x = x_j$, $j \neq i$. We want a coefficient which is unity for $x = x_i$ and zero for $x = x_j$ and whose derivative vanishes at $x = x_j$, including $i = j$.

Let us consider

$$\alpha_i(x) = (1 + b(x-x_i))\ \ell_i^2(x) \tag{2.9.3}$$

clearly $\alpha_i(x_j) = \delta_{ij}$

$$\frac{d}{dx}\,\alpha_i(x) = 2(1+b(x-x_i))\ell_i(x)\ell_i'(x) + b\ \ell_i^2(x) \tag{2.9.4}$$

$$\frac{d}{dx}\,\alpha_i(x_j) = 0 \qquad j \neq i$$

$$= 2\ell_i(x_i)\ell_i'(x_i) + b\ \ell_i^2(x_i) \qquad j = i \tag{2.9.5}$$

That is, $\frac{d}{dx}\alpha_i(x) = 0$ for all x_j if

$$b = -2\ell_i'(x_i). \tag{2.9.6}$$

Hence (2.9.3) becomes

$$\alpha_i(x) = (1-2\ell_i'(x)(x-x_i))\ \ell_i^2(x) \tag{2.9.7}$$

which is of degree 2n-1.

Also consider

$$\beta_i(x) = (x-x_i)\ \ell_i^2(x) \tag{2.9.8}$$

Then

$$\frac{d}{dx}\beta_i(x) = 2(x-x_i)\ \ell_i(x)\ \ell_i'(x) + \ell_i^2(x) \tag{2.9.9}$$

Clearly, $\frac{d}{dx}\beta_i(x_j) = \delta_{ij}$

Hence the Hermite interpolation formula may be written as

$$y(x) = \sum_{i=1}^{n} \alpha_i(x)\ y_i + \beta_i(x)\ y_i' \tag{2.9.10}$$

where $\alpha_i(x) = (1-2(x-x_i)\ell_i'(x_i))\ \ell_i^2(x)$

$$\beta_i(x) = (x-x_i)\,\ell_i^2(x) \tag{2.9.11}$$

The error term involved in the Hermite interpolation formula may be obtained directly from Lemma 2.4.1. That is,

$$E = \frac{\pi^2(x)}{(2n)!}\, f^{(2n)}(\xi) \tag{2.9.12}$$

where ξ is a point in the interval (x_1, x_n). By a proper choice of the polynomial $\pi(x)$, a high degree of accuracy can be obtained with only n ordinates as is shown in the following section.

Section 2.10 Gaussian Quadrature

Let us note that if we write

$$\pi(x) = P_n(x) \tag{2.10.1}$$

then

$$\ell_i(x) = \frac{P_n(x)}{(x-x_i)\,P_n'(x_i)} \tag{2.10.2}$$

and equations (2.9.11) become

$$\alpha_i(x) = \frac{(1-2(x-x_i)\ell_i'(x_i))}{(P_n'(x_i))^2}\,\frac{P_n^2(x)}{(x-x_i)^2}$$

$$\beta_i(x) = \frac{P_n^2(x)}{(x-x_i)(P_n'(x_i))^2} \tag{2.10.3}$$

If we now choose $P_n(x)$ to be the Legendre polynomial of order n, and x_i to be its zeroes, and if $g(x)$ is any polynomial of order 2n-1, then

$$(2.10.4)\quad \int_{-1}^{1} g(x)dx = \sum_{i=1}^{n} g_i \int_{-1}^{1} \alpha_i(x)\, dx + g_i' \int_{-1}^{1} \beta_i(x)\, dx$$

But

$$\int_{-1}^{1} \alpha_i(x)dx = \frac{1}{(P_n'(x_i))^2} \int_{-1}^{1} \{1-2(x-x_i)\ell_i'(x_i)\} \frac{P^2(x)}{(x-x_i)^2}\, dx$$

$$(2.10.5)\qquad = \frac{1}{(P_n'(x_i))^2} \frac{2}{(1-x_i^2)}$$

Also $$\int_{-1}^{1} \beta_i(x)\, dx = \frac{1}{(P_n'(x_i))^2} \int_{-1}^{1} P_n(x) \frac{P_n(x)}{(x-x_i)}\, dx$$

$$(2.10.6)\qquad = 0$$

Using (2.10.5) and(2.10.6) in (2.10.4)

$$(2.10.7)\quad \int_{-1}^{1} f(x)\, dx = \sum_{i=1}^{n} \frac{2}{(1-x_i^2)(P_n'(x_i))^2} f(x_i).$$

From the above analysis, we accomplish with n ordinates what would otherwise require 2n ordinates. In actual fact more is accomplished than a mere saving of ordinates. The

Gaussian choice of zeroes yields a convergent method of integration since the Legendre polynomials form an orthogonal set of functions. The more ordinates we take into account, the nearer we come to the true value of the integral. This is by no means true of equidistant ordinates.

However, the Gaussian method works only for the evaluation of definite integrals. Lanczos [1961] gives a counterpart of the Gaussian quadrature for the case of an indefinite integral by using Chebyshev polynomials in place of the Legendre polynomials.

CHAPTER III

LINEAR MULTI-STEP METHODS

In this chapter we present an attempt at unifying the theory on general linear multi-step methods used in solving differential systems.

The essential properties of stability, consistency and convergence of these methods are discussed in some detail. Illustrations are taken from the previous chapter.

The behaviour of discretization and round-off errors is also outlined.

Finally, techniques for optimizing the difference operator are discussed.

Section 3.1 Introduction

A large number of the methods for the numerical solution of ordinary differential equations belong to a general class of numerical formulae, namely, multi-step methods. The general linear k-step method is that of Definition 1.2.2. For simplicity of presentation, we will consider the method operating on only a single first-order equation instead of a set of such equations (1.2.9). Thus, we may write (1.2.11) as

$$\sum_{q=o}^{k} \alpha_q y_{n+q} - h \beta_q y'_{n+q} = 0 \qquad (3.1.1)$$

where α_q and β_q denote real constants which do not depend on n.

When dealing with the numerical integration of ordinary differential equations, we are faced with two basic problems. The first one is the problem of stable convergence, namely, the problem of determining necessary and sufficient conditions that, for sufficiently small length of divided intervals, approximate solutions can be obtained by numerical integration in any finite interval where the true solution of the differential system exists, and moreover, as the length of divided intervals tends to zero, these approximate solutions converge to the true solution. An answer to the problem of stable convergence was first given by Dahlquist [1956]. Since then, contributions have been made by Hull and Luxemburg [1960], Urabe [1961], and Henrici [1962], [1963].

The second of the problems is that of propagation of errors, i.e., the behaviour of the growth of errors as the steps advance. In the problem of stable convergence, the length of divided intervals is considered as a variable tending to zero, while, in the problem of propagation of errors it is considered as a fixed quantity. Associated with the problem of propagation of errors is one of estimation of errors. Estimates of errors in terms of Lipschitz constants are usually derived from the difference equations, but they are often too crude for practical use.

Section 3.2 Existence and Uniqueness of the Solution of the Difference Equation.

In order to show that the difference equation (3.1.1) has a unique solution for y_{n+k} for arbitrary values of y_n, $y_{n+1}, \ldots, y_{n+k-1}$, we write it in the form

$$(3.2.1) \qquad \sum_{q=o}^{k} \alpha_q y_{n+q} - h \beta_q f_{n+q} = 0$$

since $$y_m' = f(x_m, y_m) = f_m \quad m = 0,1\ldots$$

Furthermore, we assume that $\alpha_k \neq 0$. If $\beta_k = 0$, then y_{n+k} has a unique solution since equation (3.2.1) represents y_{n+k} explicitly as a function of $y_n, \ldots, y_{n+k-1}$. However, if $\beta_k \neq 0$, (3.2.1) represents an equation for y_{n+k} which conceivably might have several solutions or no solutions at all.

To throw some light on this problem let us write

$$(3.2.2) \qquad F(y) = h \frac{\beta_k}{\alpha_k} f(x_{n+k}, y) + c$$

where $$c = \frac{1}{\alpha_k} \{h(\beta_{k-1} f_{n+k-1} + \ldots + \beta_o f_n) - \alpha_{k-1} y_{n+k-1} - \ldots - \alpha_o y_n\}$$

and $$y = y_{n+k}$$

Then

$$(3.2.3) \qquad y = F(y)$$

Clearly, an iterative procedure of the form

$$y^{(\nu+1)} = F(y^{(\nu)}) \qquad \nu = 0,1,\ldots , \tag{3.2.4}$$

where $y^{(0)}$ is a suitable first approximation, can be set up. If $F(y)$ is defined by (3.2.2) and if $f(x,y)$ satisfies a Lipschitz condition with respect to y with a Lipschitz constant L, then

$$|F(y^*) - F(y)| \leq K\,|y^* - y| \tag{3.2.5}$$

where

$$K = \left|\frac{h\beta_k}{\alpha_k}\right| L \tag{3.2.6}$$

For sufficiently small values of h, $K < 1$.

In order to show uniqueness of the solution of (3.2.3), let us assume that y and y^* are two solutions of (3.2.3), i.e., $y = F(y)$, $y^* = F(y^*)$. By subtracting and using relation (3.2.5), we obtain

$$|y^* - y| \leq K\,|y^* - y| \tag{3.2.7}$$

which, by virtue of $|K| < 1$, is impossible unless $y^* = y$; that is, (3.2.2) can have only one solution.

To show convergence of the solution, we need only to subtract from relation (3.2.4) the relation $y^{(\nu)} = F(y^{(\nu-1)})$. It follows that

$$|y^{(\nu+1)} - y^{(\nu)}| \leq K\,|y^{(\nu)} - y^{(\nu-1)}| \tag{3.2.8}$$

Clearly, the sequence $\{y^{(\nu)}\}$ satisfies the Cauchy criterion for convergence and thus has a finite limit y. Hence, a unique solution of the difference equation exists under the specified conditions.

Section 3.3. Condition for Stability

To be a useful approximation to a differential system, the difference operator must possess certain properties. At this point it is convenient to associate with the difference equation (3.1.1) the polynomials

$$(3.3.1) \qquad \rho(\zeta) = \alpha_k \zeta^k + \alpha_{k-1} \zeta^{k-1} + \ldots + \alpha_o$$

$$(3.3.2) \qquad \sigma(\zeta) = \beta_k \zeta^k + \beta_{k-1} \zeta^{k-1} + \ldots + \beta_o$$

The following theorem imposes on $\rho(\zeta)$ a necessary condition for stability.

Theorem 3.3.1

A necessary condition for convergence of the linear multi-step method (3.1.1) is that none of the roots of the associated polynomial $\rho(\zeta)$ exceeds 1 in modulus, and that the roots of modulus 1 be simple.

Proof:

If the method is convergent, it is convergent for

the initial value problem $y' = 0$. In this particular case, equation (3.1.1) reduces to

$$\alpha_k y_{n+k} + \alpha_{k-1} y_{n+k-1} + \ldots + \alpha_o y_n = 0. \tag{3.3.3}$$

It is convenient to assume $\alpha_k = 1$ and rewrite (3.3.3) as

$$y_{n+k} = -\{\alpha_{k-1} y_{n+k-1} + \alpha_{k-2} y_{n+k-2} + \ldots + \alpha_o y_n\} \tag{3.3.4}$$

Let us assume that y_i^* are the approximate values of y_i computed by (3.1.1). It follows that

$$y_{n+k}^* = -\{\alpha_{k-1} y_{n+k-1}^* + \alpha_{k-2} y_{n+k-2}^* + \ldots + \alpha_o y_n\} + R_n \tag{3.3.5}$$

where R_n is the round-off error. If we now put

$$e_i = y_i - y_i^* \qquad i=o,1,\ldots \tag{3.3.6}$$

and assume that R_n vanishes, we can express the errors of the approximate solution by

$$e_{n+k} = -\{\alpha_{k-1} e_{n+k-1} + \alpha_{k-2} e_{n+k-2} + \ldots + \alpha_o e_n\} \tag{3.3.7}$$

Clearly, this can be written in matrix form as

$$\begin{bmatrix} e_{n+1} \\ e_{n+2} \\ \vdots \\ e_{n+k-1} \\ e_{n+k} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & 0 & 1 \\ -\alpha_o & -\alpha_1 & \ldots & -\alpha_{k-2} & -\alpha_{k-1} \end{bmatrix} \begin{bmatrix} e_n \\ e_{n+1} \\ \vdots \\ e_{n+k-2} \\ e_{n+k-1} \end{bmatrix} \tag{3.3.8}$$

If we write (3.3.8) in vector form, we have

$$\underline{e}_{n+1} = \underline{A}_o \, \underline{e}_n \tag{3.3.9}$$

from which it follows that

$$\underline{e}_n = \underline{A}_o^n \, \underline{e}_o . \tag{3.3.10}$$

From equation (3.3.8), it can readily be seen that the eigenvalues of $\underline{A}_o$ are the roots of the equation

$$\rho^*(\lambda) = \lambda^k + (\alpha_{k-1} \lambda^{k-1} + \alpha_{k-2} \lambda^{k-2} + \ldots + \alpha_1 \lambda + \alpha_o) = 0 \tag{3.3.11}$$

Obviously equations (3.3.11) and (3.3.1) have the same roots.

Since any matrix may be brought into the Jordan canonical form by a similarity transformation [Turnbull, H.W. and A.C. Aitken, [1948]], let us denote by λ_i (i= 1,2,...) the distinct roots of the equation (3.3.11) and by m_i (i=1,2,...) their multiplicities. Then the Jordan canonical form of $\underline{A}_o$ becomes the direct sum of the matrices

$$\begin{bmatrix} \lambda_i & 0 & & & 0 \\ \delta & \lambda_i & & & \\ 0 & \delta & \ddots & & \vdots \\ \vdots & & \ddots & & \\ \vdots & & & & 0 \\ 0 & \cdots & & \delta & \lambda_i \end{bmatrix} \qquad (i=1,2,\ldots) \tag{3.3.12}$$

of order m_i, where δ is an arbitrary positive number.

If $\underline{\Lambda}_o$ is the matrix of this Jordan canonical form, then

$$\underline{T}^{-1}\ \underline{A}_o\ \underline{T} = \underline{\Lambda}_o \tag{3.3.13}$$

By the substitution

$$\underline{e}_n = \underline{T}\ \underline{e}_n' , \tag{3.3.14}$$

equation (3.3.10) can be rewritten as

$$\underline{e}_n' = \underline{\Lambda}_o^n\ \underline{e}_o' \tag{3.3.15}$$

Clearly, $\underline{\Lambda}_o^n$ is the direct sum of the matrices

$$\begin{bmatrix} \lambda_i^n & 0 & \cdots & 0 \\ \binom{n}{1}\lambda_i^{n-1}\delta & \lambda_i^n & & \vdots \\ \binom{n}{2}\lambda_i^{n-2}\delta & \binom{n}{1}\lambda_i^{n-1}\delta & \ddots & 0 \\ \vdots & & \ddots & \\ & \cdots & \binom{n}{1}\lambda_i^{n-1}\delta & \lambda_i^n \end{bmatrix} \tag{3.3.16}$$

for $i = 1,2,\ldots$.

Now, in order that the difference operator be stable, $\underline{e}_n$ must be bounded for any $\underline{e}_o$ provided $|\underline{e}_n|$ is sufficiently small.

This implies that, for any $\underline{e}_0'$ whose norm is sufficiently small, $\underline{e}_n'$ determined by (3.3.15) must be bounded. Therefore, from (3.3.16) we see that for stability the following two conditions are necessary:

1. The roots of $\rho(\zeta)$ are all at most one in absolute value.
2. The roots whose absolute values are one are simple.

All the methods based on integration in the previous chapter satisfy the condition for stability. For these methods the polynomial $\rho(\zeta)$ is of the form $\zeta^k - \zeta^{k-p}$ $(1 \leq p \leq k)$ and thus has p distinct roots on the unit circle and a root of multiplicity k - p at $\zeta = 0$.

Section 3.4 Condition for Convergence

Imposing the condition of stability on a method prevents a same initial error in the computations from growing at such a rate that convergence is jeopardized. But stability alone does not guarantee convergence. A further condition of consistency must be added. From Definition 1.2.4, the condition of consistency may be expressed in terms of the polynomials $\rho(\zeta)$ and $\sigma(\zeta)$ as

$$\rho(1) = 0$$

$$\rho'(1) = \sigma(1) \; . \tag{3.4.1}$$

It can readily be verified that the methods of Chapter II which satisfy the condition of stability also satisfy the condition of consistency. In this section we shall show that the conditions of stability and consistency which are necessary for convergence of a linear multi-step method are also sufficient.

Theorem 3.4.1

A stable and consistent linear multi-step method is convergent.

Proof:

The necessary condition for convergence has been derived in the previous section. Urabe [1961] presents an elegant proof of the sufficiency condition which may be outlined as follows. If the true solution satisfies

$$(3.4.2)\quad y_{n+k} = -\{\alpha_{k-1}\, y_{n+k-1} + \ldots + \alpha_o y_n\} + h\{\beta_k\, f(x_{n+k},\, y_{n+k}) + \ldots + \beta_o\, f(x_n,\, y_n)\} + T_n$$

and the approximate solution satisfies

$$(3.4.3)\quad y^*_{n+k} = -\{\alpha_{k-1}\, y^*_{n+k-1} + \ldots + \alpha_o y^*_n\} + h\{\beta_k\, f(x_{n+k},\, y^*_{n+k}) + \ldots + \beta_o\, f(x_n,\, y^*_n)\} + R_n$$

where T_n and R_n are respectively the truncation and round-off errors, then the errors of the approximate solution satisfy

$$(3.4.4)\quad e_{n+k} = -\{\alpha_{k-1}\, e_{n+k-1} + \alpha_{k-2}\, e_{n+k-2} + \ldots + \alpha_o\, e_n\} + h\,[\beta_k\{f(x_{n+k},\, y_{n+k}) - f(x_{n+k},\, y^*_{n+k})\} + \ldots + \beta_o\,\{f(x_n,\, y_n) - f(x_n,\, y^*_n)\}] + R_n - T_n.$$

Equation (3.4.4) may now be written in matrix form as

$$\underline{e}_{n+1} = \underline{A}_o \underline{e}_n + h \underline{\psi}_n (\underline{e}_{n+1}, \underline{e}_n) + h \underline{\rho}_n \tag{3.4.5}$$

where $\underline{A}_o$ is of the form of equation (3.3.8) and $\underline{\rho}_n = o(1)$ uniformly as $|h| \to 0$.

Since $f(x,y)$ satisfies a Lipschitz condition, for certain constants k_1 and k_2,

$$| \underline{\psi}_n (\underline{e}_{n+1}, \underline{e}_n)| \leq k_1 |\underline{e}_{n+1}| + k_2 |\underline{e}_n|. \tag{3.4.6}$$

By the stability conditions we can choose $\delta > 0$ so small that

$$\delta + |\lambda_i| < 1 \tag{3.4.7}$$

for any multiple eigenvalue of $\underline{A}_o$. On the other hand, one of the eigenvalues of $\underline{A}_o$ is 1 by the consistency conditions and moreover, such an eigenvalue is simple by the stability conditions. Thus we see that the norm

$$|\underline{\Lambda}_o| = 1 \tag{3.4.8}$$

and $$\underline{T}^{-1} \underline{A}_o \underline{T} = \underline{\Lambda}_o$$

If we now put

$$\underline{e}_n = \underline{T} \underline{e}_n' , \quad \underline{\rho}_n = \underline{T} \underline{\rho}_n' \tag{3.4.9}$$

and

$$(3.4.10)\quad \underline{T}^{-1}\ \underline{\psi}_n\ (\underline{T}\ \underline{e}'_{n+1},\ \underline{T}\ \underline{e}'_n) = \underline{\psi}'_n\ (\underline{e}'_{n+1},\ \underline{e}'_n)\ ,$$

it follows readily from (3.4.6) that

$$(3.4.11)\quad |\ \underline{\psi}'_n\ (\underline{e}'_{n+1},\underline{e}'_n)| \leq k'_1\ |\ \underline{e}'_{n+1}| + k'_2\ |\underline{e}'_n| + \underline{\sigma}'_n$$

and

$$(3.4.12)\qquad |\underline{\rho}'_n| = o(1)\quad \text{uniformly as } |h| \to o\ ,$$

where

$$(3.4.13)\qquad k'_1 = |T^{-1}|\ |T|\ k_1,\quad k'_2 = |T^{-1}|\ |T|\ k_2$$

and

$$(3.4.14)\quad \underline{\sigma}'_n = o(1)\quad \text{uniformly as } |h| \to o.$$

Let us take a positive number h_o such that

$$(3.4.15)\qquad h_o\ k'_1 < 1$$

Then for any positive number ϵ, we can take a positive number h_1 $(\leq h_o)$ so that

$$(3.4.16)\qquad |\underline{\sigma}'_n| + |\underline{\rho}'_n| \leq \epsilon$$

for any h such that $|h| \leq h_1$.

Now equation (3.4.5) becomes

$$(3.4.17)\quad \underline{e}'_{n+1} = \underline{\Lambda}_o\, \underline{e}'_n + h\, \underline{\psi}'_n\,(\underline{e}'_{n+1},\, \underline{e}'_n) + h\, \underline{\rho}'_n$$

But by (3.4.8), (3.4.11) and (3.4.16), we have

$$(3.4.18)\quad |\underline{e}'_{n+1}| \le |\underline{e}'_n| + |h|\,(k'_1\,|\underline{e}'_{n+1}| + k'_2\,|\underline{e}'_n|) + |h|\epsilon$$

which, due to (3.4.15), may be written as

$$(3.4.19)\quad |\underline{e}'_{n+1}| \le \frac{1 + |h|\,k'_2}{1 - |h|\,k'_1}\,|\underline{e}'_n| + \frac{|h|}{1 - |h|\,k'_1}\,\epsilon$$

But it is easily verified that for $|h| \le h_o$,

$$\frac{1 + |h|\,k'_2}{1 - |h|\,k'_1} \le 1 + \frac{k'_2 + k'_2}{1 - h_o\,k'_1}\,|h|$$

$$(3.4.20)\quad \frac{|h|}{1 - |h|\,k'_1} \le \frac{|h|}{1 - h_o\,k'_1}$$

Therefore, from (3.4.19), follows

$$(3.4.21)\quad |\underline{e}'_{n+1}| \le \left(1 + \frac{k'_1 + k'_2}{1 - h_o k'_1}\,|h|\right)|\underline{e}'_n| + \frac{|h|}{1 - h_o k_1}\,\epsilon$$

Then, by induction, we have

$$(3.4.22)\quad |\underline{e}_n'| \leq \left(1 + \frac{k_1' + k_2'}{1 - h_o k_1'} |h|\right)^n |\underline{e}_o'| + \frac{1}{k_1' + k_2'}$$

$$\left\{\left(1 + \frac{k_1' + k_2'}{1 - h_o k_1'} |h|\right)^n - 1\right\} \epsilon$$

Clearly, this is a monotonically increasing function which for large n may be written as

$$(3.4.23)\quad |\underline{e}_n'| \leq \left(\exp \frac{k_1' + k_2'}{1 - h_o k_1'} L\right) |\underline{e}_o'| + \frac{1}{k_1' + k_2'}$$

$$\left\{\left(\exp \frac{k_1' + k_2'}{1 - h_o k_1'} L\right) - 1\right\} \epsilon$$

where $n|h| \leq L$. The inequalities (3.4.23) imply that $|\underline{e}_n'| \to 0$ uniformly as $|\underline{e}_o'|$ and $|h|$ tend to zero. This, in turn, implies that $|\underline{e}_n| \to 0$ uniformly as $|\underline{e}_o|$ and $|h|$ tend to zero. Thus, the theorem is proven.

Section 3.5 Propagation of Errors

As can be seen from the previous section, any attempt to achieve an error bound on the general multi-step formula results in unwieldy expressions. However, a crude error approximation may be associated with the order of the difference approximation. That is, we associate with (3.1.1) the difference operator

$$(3.5.1)\quad L\,[y(x);\ h] = \alpha_k\, y(x+kh) + \alpha_{k-1}\, y(x+(k-1)h) + \ldots + \alpha_o\, y(x) - h\{\beta_k\, y'(x+kh) + \ldots + \beta_o\, y'(x)\}$$

We may now expand $L\,[y(x);\ h]$ in powers of h, and the expansion can be taken as far as we please. In view of the formulae

$$y(x+mh) = y(x) + mh\, y'(x) + \frac{m^2h^2}{2}\, y''(x) + \ldots$$

$$(3.5.2)\quad h\, y'(x+mh) = h\, y'(x) + mh^2\, y''(x) + \ldots$$

it turns out that

$$(3.5.3)\quad L\,[y(x);\ h] = C_o\, y(x) + C_1 h\, y'(x) + \ldots + C_q h^q y^{(q)}(x) + \ldots$$

where the coefficients C_q $(q=0,1,\ldots)$ are constants which do not depend on the choice of the function $y(x)$:

$$C_o = \alpha_o + \alpha_1 + \ldots + \alpha_k$$

$$C_1 = \alpha_1 + 2\alpha_2 + \ldots + k\,\alpha_k - (\beta_o + \beta_1 + \ldots + \beta_k)$$

$$(3.5.4)\quad C_q = \frac{1}{q!}\,(\alpha_1 + 2^q\alpha_2 + \ldots + k^q\,\alpha_k) - \frac{1}{(q+1)!}$$

$$(\beta_1 + 2^{q-1}\,\beta_2 + \ldots + k^{q-1}\,\beta_k)$$

The difference operator (3.5.1) is said to be of order p if $C_o = C_1 = \ldots = C_p = 0$ but $C_{p+1} \neq 0$. Then

$$(3.5.5) \quad L\,[y(x);\ h] = C_{p+1}\ h^{p+1}\ y^{(p+1)}(x) + O(h^{p+2}).$$

The constant C_{p+1} can be made as small numerically as we please by multiplying (3.1.1) by a suitable constant. However, it happens that the constant

$$(3.5.6) \qquad C = \frac{C_{p+1}}{\beta_o + \beta_1 + \ldots + \beta_k}$$

which is invariant, may be used as a measure of accuracy of the method. This error constant is defined for all convergent methods.

More refined expressions for both the discretization errors and the round-off errors may be developed. Henrici [1962] shows that the discretization error consists of essentially two parts. The first part which may be called the genuine discretization error is represented by a term proportional to C. The second part owes its existence to the fact that the starting error is not zero. However, it tends to zero as $h \to 0$. Henrici [1962] shows that if, instead of satisfying the difference equation (3.1.1) exactly, the calculated y_n^* satisfy

$$(3.5.7) \quad \alpha_k\ y^*_{n+k} + \alpha_{k-1}\ y^*_{n+k-1} + \ldots + \alpha_o\ y^*_n = h\ \{\beta_k\ f(x_{n+k},\ y^*_{n+k}) + \ldots + \beta_o\ f(x_n,\ y^*_{n+k})\} + \epsilon_{n+k},$$

then the round-off error is $O(\frac{\epsilon}{h})$ where

$$|\epsilon_{n+k}| \leq \epsilon \qquad (n= 0,1,\ldots).$$

Section 3.6 Optimizing the Difference Operator

If we count the number of parameters at our disposal, we see that the operator (3.1.1) has 2k+2 coefficients (k+1 coefficients α_μ and β_μ each). Since two conditions of consistency must also be satisfied, this suggests that the highest order, p, that can be obtained is 2k. However, Henrici [1962] shows that the order of a stable operator, whose stepnumber k is odd, cannot exceed k+1 and if k is even the largest p is k+2.

In recent years various techniques have been employed in an attempt to increase the order of the approximation. Urabe [1960] suggests an extension of (3.1.1) which includes a term $h\,\beta_{k+1}\,f_{n+k+1}$. That is, a predictor equation for y_{n+k+1} and a corrector equation for y_{n+k} are solved simultaneously. Gragg and Stetter [1964] suggest a scheme which includes the value of the derivative at one non-step point x_{n+r} where r is usually non-integral. The values of f(x,y) at the point x_{n+r} are obtained from an explicit predictor. The order of these predictor-corrector methods is not subject to the restrictions of the stable difference operator.

CHAPTER IV

A MATRIX FORMULATION OF THE INITIAL VALUE PROBLEM

In this chapter we consider a matrix formulation of an nth order linear differential equation. The matrix operators which produce successive differences of a univariate table are outlined.

A matrix-factorization procedure is presented for linear equations with nearly constant coefficients.

An algorithm for the reduction of the order of a matrix approximation when a solution is known is also presented.

Section 4.1 Introduction

The systematic use of matrix arrays in the formulation of difference operations and difference equations is a recent development. Not only does this technique provide a compact notation for the essential elements of a computing procedure but it often points out computational shortcuts.

In the representation of the matrix operators we use three simple matrices: the unit matrix $\underline{I}$ and the auxiliary unit matrices $\underline{S}$ and $\underline{S}^*$ (the transpose of $\underline{S}$). The only non-zero elements of $\underline{S}$ are units in the first superdiagonal.

The familiar difference operators can be represented by linear combinations of the matrices $\underline{I}$, $\underline{S}$ and $\underline{S}^*$. Let us consider a sequence of function values which we array in a column $\underline{y}$. That is,

$$(4.1.1) \qquad \underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix}$$

Then the backward differences are given by

$$(4.1.2) \qquad \nabla \underline{y} = (\underline{I} - \underline{S}^*)\, \underline{y} = \begin{bmatrix} y_1 \\ y_2 - y_1 \\ y_3 - y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n - y_{n-1} \end{bmatrix}$$

which contains one redundant element, y_1. Operating again with $(\underline{I} - \underline{S}^*)$, we obtain

$$(4.1.3) \qquad \nabla^2 \underline{y} = (\underline{I} - \underline{S}^*)^2\, \underline{y}$$

i.e., a column of second differences with two redundant elements. Noting, however, that $(\underline{I} - \underline{S}^*)$ possesses an inverse, we have

$$(4.1.4) \qquad y = (\underline{I} - \underline{S}^*)^{-1}\, \nabla \underline{y}.$$

The inverse $(\underline{I} - \underline{S}^*)^{-1}$ is a lower-triangular matrix whose elements, e_{ij}, are $e_{ij}=1$, $j \leq i$; $e_{ij}=0$, $j>i$. It can readily be shown that

$(\underline{I} - \underline{S}^{*})^{-1}$ is a summation operator.

Section 4.2 Matrix Formulation of a Linear Initial Value Problem

At this point we return to the original problem of Chapter I and consider a differential equation of the form (1.2.1). That is,

$$(4.2.1) \quad y^{(n)}(x) = f(y^{(n-1)}(x),\ y^{(n-2)}(x),\ldots,y'(x),\ y(x),\ x)$$

Furthermore, we assume that $y^{(n)}(x)$ is a linear function of $y(x)$ and its derivatives; that is,

$$(4.2.2) \quad y^{(n)}(x) = \sum_{i=o}^{n-1} a_i\, y^{(i)}(x) + c(x)$$

where a_i are constants. We follow the current practice of replacing the differential operators by difference expansions truncated at a suitable order of difference. If the equation to be solved is linear and formally soluble for the highest derivative,the computing routines for this type of initial value problem are usually straightforward and in general do not make use of any simplifying features in the form of the equation itself.

A simple example will serve to illustrate the main point of this section. Let us suppose that we wish to determine the solutions of the differential system

$$\frac{d^2y}{dx^2} - y = 0, \quad y(0) = 1. \tag{4.2.3}$$

If we approximate to the differential equation by a second-order central difference equation, we obtain

$$\delta^2\left(1 - \frac{1}{12}\delta^2 + \ldots\right)y_i - h^2 y_i = 0. \tag{4.2.4}$$

Truncating the series after the second difference, we have

$$\delta^2 y_i - h^2 \left(1 - \frac{1}{12}\delta^2\right)^{-1} y_i = 0$$

Or by simplifying,

$$\left(1 - \frac{h}{12}\right)\delta^2 y_i - h^2 y_i = 0. \tag{4.2.5}$$

But since

$$\delta = E^{\frac{1}{2}} - E^{-\frac{1}{2}} \tag{4.2.6}$$

where E is the forward shift operator, the difference equation (4.2.5) can be written as

$$\left(E^2 - \frac{24 + 10h^2}{12 - h^2} E + 1\right) y_i = 0. \tag{4.2.7}$$

Clearly, this is equivalent to

(4.2.8)
$$y_{i+1} - \rho_1 y_i = 0$$
$$y_{i+1} - \rho_2 y_i = 0$$

where ρ_1, ρ_2 are the roots of the quadratic operator in (4.2.7). If we now choose initial conditions that are appropriate to the system of difference equations (4.2.8), two approximations to (4.2.3) can be determined.

Each of the equations of (4.2.8) may be considered as an approximation to a first-order initial value problem. In fact, it is obvious that the basic operator for initial value difference systems is of the form $(\underline{I} - \underline{S}^* \underline{A})$ where $\underline{A}$ is a diagonal matrix. Or in general, a first-order problem may be written as

(4.2.9)
$$(\underline{I} - \underline{S}^* \underline{A})\, \underline{y} = \underline{f}$$

where $\underline{f}$ represents the initial condition, i.e., $\underline{f}$ has a single non-zero element, f_1. From equation (4.2.7) it is clear that a second-order problem is of the form

(4.2.10)
$$(\underline{I} - \underline{S}^* \underline{A} + \underline{S}^{*2} \underline{B})\, \underline{y} = \underline{f}$$

where $\underline{f}$ now has two non-zero elements, f_1 and f_2. Obviously, this notation can easily be extended to a representation of an nth order difference approximation.

Section 4.3 Solution of Linear Equations with Nearly Constant Coefficients

For simplicity, let us again consider the matrix representation of a second-order system, (4.2.10). It will be convenient to express the operator as a product of two factors; we write

$$(\underline{I} - \underline{S}^{*}\underline{A} + \underline{S}^{*2}\underline{B}) = (\underline{I} - \underline{S}^{*}\underline{C})(\underline{I} - \underline{S}^{*}\underline{D}) \tag{4.3.1}$$

where $\underline{A}$, $\underline{B}$, $\underline{C}$, $\underline{D}$ are diagonal matrices. Denoting the non-zero elements of $\underline{A}$, $\underline{B}$, $\underline{C}$, $\underline{D}$ by a_i, b_i, c_i, d_i $(i=1,2,\ldots)$ we have the recurrence relations

$$c_i + d_i = a_i$$

$$c_{i+1}\, d_i = b_i \tag{4.3.2}$$

Obviously, if a pair of starting values c_1, d_1 are known, all the elements of $\underline{C}$ and $\underline{D}$ can be determined.

Now the form of equation (4.2.7) suggests that we approximate the difference operator (4.3.1) by successive quadratics

$$\rho^2 - a_i\rho + b_i = 0 \tag{4.3.3}$$

If we choose c_1, d_1 to be the roots ρ_1, ρ_2 of the quadratic (4.3.3) at $i=1$, we can determine all the elements of $\underline{C}$ and $\underline{D}$.

However, if c_i and d_i are the roots of (4.3.3) at each step, we require that

$$c_i + d_i = a_i \tag{4.3.4}$$

$$c_i d_i = b_i$$

In order that equations (4.3.4) be consistent with (4.3.2), we require that

$$c_{i+1} = c_i \tag{4.3.5}$$

This is possible only if the coefficients of the difference equation are nearly constant.

We now ignore the initial values on the right of (4.2.10) and consider instead two specialized problems. By writing

$$\underline{y} = (\underline{I} - \underline{S}^* \underline{D})^{-1} \underline{w} \tag{4.3.6}$$

in (4.2.10), we have

$$(\underline{I} - \underline{S}^* \underline{C})\underline{w} = \underline{f}. \tag{4.3.7}$$

If we choose the components of $\underline{f}$ in (4.3.7) to be

1, $-c_1$, 0, 0, ..., the components of $\underline{w}$ are 1, 0, 0, 0, ... and $\underline{y}$ is determined by (4.3.6). It is easily verified that the choice of the initial value vector $\underline{f}$ in (4.3.7) determines the initial values of $\underline{y}$ to be 1, d_1. Clearly, by forcing initial values on the difference system, one solution can be isolated by means of the first factorization (4.3.6). The second factorization (4.3.7) permits the isolation of the second solution. The solution of the original initial value problem can then be expressed as the sum of two simple initial value problems. Obviously, this procedure can be extended to an nth order problem.

An algorithm for isolating the particular solutions of a linear differential equation with nearly constant coefficients may be carried out in the following steps:

(i) Replace the differential operators by difference expansions truncated at a suitable order of difference.

(ii) Express the difference system in a matrix notation of the form (4.2.10).

(iii) Factor the difference matrix into a product of factors $(\underline{I} - \underline{S}^{*}\underline{P}_j)$, $(j=1,\ldots,n)$.

(iv) By choosing appropriate initial conditions separate the problem into n first-order problems.

In factoring the difference matrix in step (iii), the starting values for the recurrence relations in the form of

(4.3.2) are the roots of the difference polynomial in the form (4.3.3). In practice, this polynomial is often ill-conditioned. However, Wilkinson [1959] outlines modifications to Bairstow's method in order to determine the roots of an ill-conditioned polynomial to the desired accuracy. Such a method was used in the illustration which follows.

Let us find the solutions of the third-order differential equation

$$y''' - 7y' + 6 = 0. \tag{4.3.8}$$

It is easily verified that the analytic solution of (4.3.8) is

$$y = c_1 e^{x} + c_2 e^{2x} + c_3 e^{-3x}. \tag{4.3.9}$$

If we now approximate the differential operators of (4.3.8) by forward differences, we have

$$\left[\Delta^3\left(1-\tfrac{1}{2}\Delta + \tfrac{1}{3}\Delta^2 - \tfrac{1}{4}\Delta^3 + \ldots\right)^3 - 7h^2\,\Delta\left(1-\tfrac{1}{2}\Delta + \tfrac{1}{3}\Delta^2 - \tfrac{1}{4}\Delta^3 + \ldots\right) + 6h^3\right] y = 0 . \tag{4.3.10}$$

Truncating the series after the third difference and simplifying, we have

$$\left[\left(1-\tfrac{7}{12}h^2\right)\Delta^3 + \left(3h^3 - 7h^2\right)\Delta^2 + \left(9h^3 - 7h^2\right)\Delta + 6h^3\right] y = 0. \tag{4.3.11}$$

Since $\Delta = E - 1$, equation (4.3.11) may be written as

$$(4.3.12)\quad \left[(1-\frac{7}{12}h^2)E^3 + (3h^3 - \frac{21}{4}h^2-3)E^2 + (3h^3 + \frac{21}{4}h^2+3)E + (\frac{7}{12}h^2 - 1)\right] y = 0.$$

If we choose $h = \frac{1}{8}$, the difference polynomial becomes

$$(4.3.13)\quad \left[E^3 - 3.10446779\ E^2 + 3.11629434\ E - 1\right] y = 0.$$

Since the coefficients are constants, the elements in the corresponding matrix formulation, $(\underline{I} - \underline{S}^*\underline{A} + \underline{S}^{*2}\underline{B} - \underline{S}^{*3}\underline{C})$, are also constants. Furthermore, the roots of the difference polynomial (4.3.13) are

$$(4.3.14)\quad \begin{aligned} \rho_1 &= .6872\ 8398 \\ \rho_2 &= 1.1331\ 4810 \\ \rho_3 &= 1.2840\ 3560 \end{aligned}$$

The factors $(\underline{I} - \underline{S}^*\underline{D})$, $(\underline{I} - \underline{S}^*\underline{E})$ and $(\underline{I} - \underline{S}^*\underline{F})$ can be written at once: $d_i = \rho_1$; $e_i = \rho_2$ and $f_i = \rho_3$. The solutions of the three reduced problems are given in Table I, along with the corresponding values of the exact solution.

The values obtained in the approximations are quite accurate despite the fact that forward difference expressions were used to approximate the differential operators. Since

central difference approximations converge more rapidly than do forward difference approximations, they would provide a more accurate approximation to the differential equation. The price to be paid is the difficulty of expressing the resulting difference system in the required matrix form.

TABLE I

SOLUTION OF

$$y''' - 7y' + 6y = 0$$

$$y = (\underline{I} - \underline{S}*\underline{D})^{-1}\underline{f}$$

x	e^{-3x}	y(x)	$10^8(e^{-3x} - y)$
0.0	1.0000 0000	1.0000 0000	0
0.125	.6872 8927	.6872 8398	529
0.250	.4723 6655	.4723 5926	729
0.375	.3246 5246	.3246 4495	751
0.500	.2231 3016	.2231 2327	689
0.625	.1533 5496	.1533 4905	591
0.750	.1053 9922	.1053 9445	477

$$y = (\underline{I} - \underline{S}*\underline{E})^{-1}\underline{f}$$

x	e^{x}	y(x)	$10^8(e^{x} - y)$
0.0	1.0000 0000	1.0000 0000	0
0.125	1.1331 4845	1.1331 4810	35
0.250	1.2840 2541	1.2840 2461	- 20
0.375	1.4549 9141	1.4549 9005	136
0.500	1.6487 2127	1.6487 1921	206
0.625	1.8682 4595	1.8682 4304	291
0.750	2.1170 0001	2.1169 9605	396

$$y = (\underline{I} - \underline{S}*\underline{F})^{-1}\underline{f}$$

x	e^{2x}	y(x)	$10^8(e^{2x} - y)$
0.0	1.0000 0000	1.0000 0000	0
0.125	1.2840 2541	1.2840 3560	- 1019
0.250	1.6487 2127	1.6487 4742	- 2615
0.375	2.1170 0001	2.1170 5038	- 5037
0.500	2.7182 8182	2.7183 6806	- 8624
0.625	3.4903 4295	3.4904 8136	- 13841
0.750	4.4816 8907	4.4819 0232	- 21325

Section 4.4 The Reduction of the Order of a Difference Approximation When a Solution is Known.

In the previous section we considered the complete numerical solution of a very special class of differential equations. As may be suspected a single algorithm that will cope with any general case is difficult to develop. In this section, we develop an algorithm that reduces the order of a difference approximation when a solution is known.

For simplicity let us again consider a second-order difference equation of the form

$$u_{n+2} + a\, u_{n+1} + b\, u_n = 0 \tag{4.4.1}$$

where a, b need not be constants. Also, let us suppose that v is a solution of (4.4.1). Then we are looking for a solution of the type vw; that is,

$$w_{n+2}\, v_{n+2} + a\, v_{n+1}\, w_{n+1} + b\, v_n w_n = 0. \tag{4.4.2}$$

Equation (4.4.2) may be written as

$$(v_{n+2} + a\, v_{n+1} + b\, v_n)\, w_{n+2} + a\, v_{n+1}\,(w_{n+1} - w_{n+2}) + b\, v_n (w_n - w_{n+2}) = 0. \tag{4.4.3}$$

Since v is a solution of (4.4.1), equation (4.4.3) becomes

$$(4.4.4) \quad a\, v_{n+1}(w_{n+1} - w_{n+2}) + b\, v_n(w_n - w_{n+1} + w_{n+1} - w_{n+2}) = 0$$

Clearly, this reduces to

$$(4.4.5) \quad (a\, v_{n+1} + b\, v_n)\, z_{n+1} + b\, v_n\, z_n = 0$$

where $z_n = \Delta w_n$. Equation (4.4.5) is obviously a first-order approximation.

If a solution of the reduced equation (4.4.5) can be determined, a second solution of the original problem (4.4.1) can be calculated by working backwards. That is,

$$(4.4.6) \quad w_n = \sum_{i=1}^{n} z_i \,,$$

and a second solution is $v_n w_n$ which includes a linear contribution of the known solution v. The constants can be determined from the initial conditions of the problem.

Clearly, this method can be applied to a nth order approximation. If at each step a solution can be obtained for the reduced equation, this procedure deflates an nth order approximation. The accuracy of the final result depends largely on the accuracy of the solution of the reduced equation at each step.

CHAPTER V

CONCLUDING REMARKS

In the preceding chapters, we have reviewed numerical techniques for the solution of a special, but important, class of differential equations, namely, the initial value problem. Certain difficulties - such as stability and convergence - which confront us in our attempts to obtain useful numerical results have been discussed in some detail. Error propagation has been treated very briefly, but at sufficient length to show that the problem is not an easy one. In Chapter IV, we examined the isolation of each of the solutions of an ordinary differential equation subject to initial conditions.

The approximation to a continuous system by discrete functions is a field of considerable difficulty. Difference approximation, however, remains the most widely applicable technique available. Matrix formulations frequently enable us to devise concise and easily-checked algorithms for the solutions of such difference systems. Furthermore, the concept of eigen-functions as applied to differential and difference operators emerges naturally.

The matrix-factorization algorithm developed in Chapter IV provides an efficient method for determining all the solutions of a linear differential equation with nearly constant coefficients. However, an extension to a non-linear system would require the explicit factorization of the difference polynomial at each step,

which is time-consuming if the order of the differential system is large. A further problem which arises in the suggested procedure for deflating an nth order difference approximation is one of determining a solution of the reduced approximation at each step.

Throughout our investigations, we have considered only well-behaved differential systems. The study of the effect of singularities on numerical results is a difficult problem and little has been done in this field.

The numerical study of differential equations is a conglomeration of special devices. It is only recently that intensive study of the subject has resulted in some unification; but much work remains to be done.

BIBLIOGRAPHY

ABRAMOWITZ, M. and I. A. STEGUN, (ed.), 1964. Handbook of Mathematical Functions, 1st edition, National Bureau of Standards, Washington, D.C.

BELLMAN, R. and K. L. COOKE, 1963. Differential-Difference Equations, 1st edition, Academic Press, New York.

BICKLEY, W. G., 1948. "Difference and Associated Operators, With Some Applications", Journal of Mathematics and Physics, Vol. 27, No.3, Oct.

BICKLEY, W. G. and J. McNAMEE, 1960. "Matrix and Other Direct Methods for the Solution of Systems of Linear Difference Equations", Phil. Trans. Roy. Soc., Vol. 252, Jan., pp 69-131.

BUCKINGHAM, R. A., 1962. Numerical Methods, 2nd edition, Pitman, London.

DAHLQUIST, G., 1956. "Convergence and Stability in the Numerical Integration of Ordinary Differential Equations", Math. Scand., Vol. 4, pp. 33-53.

DAVIS, P.J., 1963. Interpolation and Approximation, 1st edition, Blaisdell Publishing Co., New York.

GRAGG, W. B. and H. J. STETTER, 1964. "Generalized Multi-step Predictor-Corrector Methods", ACM, Vol. 11, No. 2, April, pp. 188-209.

HAMMING, R. W., 1962. Numerical Methods for Scientists and Engineers, 1st edition, McGraw-Hill, New York.

HENRICI, P., 1962. Discrete Variable Methods in Ordinary Differential Equations, 1st edition, John Wiley & Sons, New York.

1963. Error Propagation for Difference Methods, 1st edition, John Wiley & Sons, New York.

HILDEBRAND, F.B., 1956. Introduction to Numerical Analysis, 1st edition, McGraw-Hill, New York.

HULL, T. E. and W. A. T. LUXEMBURG, 1960. "Numerical Methods and Existence Theorems for Ordinary Differential Equations", Numerische Mathematik, Vol. 2, pp. 30-41.

HULL, T.E. and A.C.R. NEWBERRY, 1962. "Corrector Formulas for Multi-step Integration Methods", J.Soc. Indust. Appl. Math., Vol. 10, No. 2, June, pp. 351-369.

HULL, T.E. and A. L. CREEMER, 1963. "Efficiency of Predictor-Corrector Methods", ACM , Vol. 10, No. 3, July, pp. 291-301.

IVERSON, K.E., 1962. A Programming Language, 1st edition, John Wiley & Sons, New York.

LANCZOS, C., 1961. Linear Differential Operators, 1st edition, D. Van Nostrand Co., Toronto.

MAILLOUX, B.J.,1963. Numerical Solution of Differential Equations, Master's Thesis, University of Alberta, Sept.

MILNE, W. E., 1953. Numerical Solution of Differential Equations, 1st edition, John Wiley & Sons, New York.

MILNE-THOMSON, L.M., 1933. The Calculus of Finite Differences, 1st edition, MacMillan and Co., London.

MURRAY, W.A., 1962. The Numerical Solution of Partial Differential Equations by Closed Difference Methods, Master's Thesis, University of Alberta, Oct.

NATIONAL PHYSICAL LABORATORY, 1961. Notes on Applied Science No. 16: Modern Computing Methods, 2nd edition, H.M. Stationery Office, London.

TODD, J., (ed.), 1962. Survey of Numerical Analysis, 1st edition, McGraw-Hill, New York.

TURNBULL, H.W. and A.C. AITKEN, 1948. An Introduction to the Theory of Canonical Matrices, 2nd edition, Blackie & Son, London.

URABE, M., 1956. "Convergence of Numerical Iteration in Solution of Equations", J. Sci. Hiroshima Univ., Series A, Vol. 19, No. 3, pp. 479-489.

1960. "Periodic Solutions of Van Der Pol's Equation with Damping Coefficient λ = 2-10", J. Sci. Hiroshima Univ., Series A, Vol. 23, pp. 325-366.

1961. "Theory of Errors in Numerical Integration of Ordinary Differential Equations", J. Sci. Hiroshima Univ., Series A-I, Vol. 25, No. 1, pp. 3-62.

WILKINSON, J.H., 1959. "The Evaluation of the Zeroes of Ill-conditioned Polynomials", Numerische Mathematik, Vol. 1, pp. 150-180.

APPENDIX

Solution of $y''-(1-\frac{1}{4x^2})\,y = 0$

The analytic solution of the modified Bessel equation

$$y'' - \left(1 - \frac{1}{4x^2}\right) y = 0$$

is

$$y = x^{1/2} \{A\, K_o(x) + B\, I_o(x)\}$$

We attempt to determine the decreasing solution $x^{1/2}K_o(x)$ in the range $x = 2.0$ to $x = 16.0$, given the initial data

$$y(2) = .1610\ 7025 = (2)^{1/2}K_o(2)$$

$$y(2.4) = .1087\ 8023 = (2.4)^{1/2}K_o(2.4).$$

The tabular interval used is $h = 0.4$ which in practice would be considered large.

If we approximate to the differential equation by a second-order central difference equation, we obtain

$$\delta^2\left(1 - \frac{1}{12}\delta^2\right)y_i - h^2 q_i y_i = 0$$

where

$$q_i = 1 - \frac{1}{4x_i^2}$$

Assuming that $Eq_i y_i = q_i E y_i$

and that $\left(1 - \frac{1}{12}\delta^2\right) = \left(1 + \frac{1}{12}\delta^2\right)$,

this may be written as

$$\left(E^2 - \left(\frac{24 + 10\, q_i h^2}{12 - q_i h^2}\right) E + 1\right) y_i = 0.$$

The non-zero elements of the component matrices encountered in the computation are tabulated in Table II. Since q_i varies least for large x, a backward factorization is used. That is, we let

$$c_n = \rho'_n \ , \quad d_n = \rho''_n$$

and apply the recurrence relations (4.3.2) backwards. The solution Y_M obtained from the matrix method is also tabulated in Table II.

Let Y_T be $x^{1/2}K_o(x)$ and Y_M be the solution obtained from the matrix method. Y_{N1} and Y_{N2} are both solutions obtained using Numerov's method, Y_{N1} and Y_{N2} agreeing with Y_T and Y_M respectively at both x = 2 and x = 2.4. In Table III we tabulate x, Y_T, Y_T-Y_M, Y_T-Y_{N1} and Y_T-Y_{N2}.

As expected, Y_T-Y_{N1} while initially very small, soon grows unreasonably large; $(Y_T-Y_{N1})/Y_T$, then, grows even faster. On the other hand, Y_T-Y_M, though initially larger than Y_T-Y_{N1}, decreases in such a manner that $(Y_T-Y_M)/Y_T$ stays relatively constant at about 5×10^{-3}. It is not surprising, then, that Y_T-Y_{N2} grows even faster than Y_T-Y_{N1}.

TABLE II

SOLUTION OF

$$y'' - \left(1 - \frac{1}{4x^2}\right) y = 0$$

X	A	C	D	Y_M
2.0	2.15189873	1.47671640	.67518233	.16107025
2.4	2.15503294	1.48108140	.67395153	.10855354
2.8	2.15692355	1.48378621	.67313733	.07307144
3.2	2.15815004	1.48558095	.67256999	.04914565
3.6	2.15899258	1.48683408	.67215849	.03303367
4.0	2.15959469	1.48774433	.67185034	.02219368
4.4	2.16004019	1.48842671	.67161348	.01490557
4.8	2.16037907	1.48895163	.67142743	.01000801
5.2	2.16064281	1.48936421	.67127859	.00671816
5.6	2.16085208	1.48969445	.67115762	.00450894
6.0	2.16102092	1.48996295	.67105796	.00302576
6.4	2.16115910	1.49018423	.67097487	.00203021
6.8	2.16127363	1.49036876	.67090486	.00136207
7.2	2.16136961	1.49052428	.67084532	.00091374
7.6	2.16145083	1.49065657	.67079426	.00061293
8.0	2.16152019	1.49077005	.67075013	.00041112
8.4	2.16157986	1.49086813	.67071173	.00027574
8.8	2.16163160	1.49095247	.67067812	.00018493
9.2	2.16167673	1.49102820	.67064852	.00012402
9.6	2.16171634	1.49109401	.67062232	.00008317
10.0	2.16175123	1.49115224	.67059902	.00005577
10.4	2.16178228	1.49120407	.67057820	.00003740
10.8	2.16180960	1.49125036	.67055953	.00002508
11.2	2.16183461	1.49129189	.67054272	.00001681
11.6	2.16185681	1.49132928	.67052753	.00001127
12.0	2.16187683	1.49136307	.67051375	.00000756
12.4	2.16189494	1.49139371	.67050123	.00000506
12.8	2.16191139	1.49142156	.67048981	.00000339
13.2	2.16192635	1.49144696	.67047938	.00000227
13.6	2.16194001	1.49147017	.67046984	.00000152
14.0	2.16195252	1.49149138	.67046114	.00000102
14.4	2.16196401	1.49151074	.67045326	.00000068
14.8	2.16197457	1.49152826	.67044631	.00000046
15.2	2.16198432	1.49154374	.67044057	.00000030
15.6	2.16199332	1.49155649	.67043683	.00000020
16.0	2.16200166	1.49156482	.67043683	.00000013

TABLE III

COMPARISON OF SOLUTIONS

X	Y_T	Y_T-Y_M	Y_T-Y_{N1}	Y_T-Y_{N2}
2.0	.16107025	0.00000000	0.00000000	0.00000000
2.4	.10878023	.00022669	0.00000000	.00022669
2.8	.07332485	.00025341	.00000025	.00048886
3.2	.04936343	.00021777	.00000005	.00082728
3.6	.03320323	.00016956	-.00000068	.00129605
4.0	.02231935	.00012566	-.00000196	.00197047
4.4	.01499610	.00009052	-.00000388	.00295909
4.8	.01007206	.00006404	-.00000666	.00442110
5.2	.00676294	.00004477	-.00001067	.00659208
5.6	.00453998	.00003103	-.00001650	.00982198
6.0	.00304715	.00002138	-.00002506	.01463179
6.4	.00204488	.00001466	-.00003770	.02179768
6.8	.00137210	.00001002	-.00005645	.03247657
7.2	.00092057	.00000683	-.00008434	.04839323
7.6	.00061758	.00000464	-.00012585	.07211930
8.0	.00041428	.00000315	-.00018769	.10748937
8.4	.00027788	.00000213	-.00027986	.16022151
8.8	.00018638	.00000144	-.00041725	.23884271
9.2	.00012500	.00000098	-.00062208	.35606908
9.6	.00008383	.00000066	-.00092750	.53086437
10.0	.00005622	.00000044	-.00138291	.79151021
10.4	.00003770	.00000030	-.00206202	1.18018534
10.8	.00002528	.00000020	-.00307473	1.75979552
11.2	.00001695	.00000013	-.00458496	2.62416071
11.6	.00001136	.00000009	-.00683721	3.91320950
12.0	.00000762	.00000006	-.01019612	5.83564274
12.4	.00000511	.00000004	-.01520555	8.70273784
12.8	.00000342	.00000002	-.02267670	12.97877104
13.2	.00000229	.00000001	-.03381949	19.35622705
13.6	.00000154	.00000001	-.05043858	28.86798274
14.0	.00000103	0.00000000	-.07522574	43.05464259
14.4	.00000069	0.00000000	-.11219594	64.21414168
14.8	.00000046	0.00000000	-.16733793	95.77406344
15.2	.00000031	0.00000000	-.24958451	142.84700740
15.6	.00000020	0.00000000	-.37226001	213.05900820
16.0	.00000013	0.00000000	-.55523934	317.78525820

B29824